DOCUMENT RESUME

ED 336 074 IR 015 083

AUTHOR Lee, Mi Ok C.
TITLE Guided Instruction with Logo Programming and the Development of Cognitive Monitoring Strategies among College Students.
PUB DATE 91
NOTE 28p.; Paper presented at the Annual Conference of the American Educational Research Association (Chicago, IL, April 3-7, 1991). Cover title differs.
PUB TYPE Information Analyses (070) -- Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Cognitive Processes; *College Students; Computer Assisted Instruction; Discovery Learning; Educational Strategies; Higher Education; *Learner Controlled Instruction; Problem Solving; *Programed Instruction; Programing; *Thinking Skills; Transfer of Training
IDENTIFIERS *Guided Discovery Approach; *Logo Programing Language

ABSTRACT

This study examined the effects of guided programmed instruction with Logo programming on the development of cognitive processes and thinking skills among college students. Logo programming has been advocated as a powerful tool to help students develop higher order thinking, an awareness of their thinking processes, and problem solving skills; however, it is suggested that Logo programming alone is not enough. Students must develop cognitive monitoring strategies, i.e., strategies that enable them to be independent learners and to transfer learning or training from one situation to another. Guided Logo instruction in this study involved three pedagogical elements: (1) Logo programming was selected as a learning tool to teach cognitive monitoring strategies; (2) an explicit model of cognitive monitoring processes was included in the Logo programming; and (3) explicit teacher mediated instruction of cognitive monitoring activities was facilitated in a Logo programming environment as well as outside of the programming domain. The results of the study demonstrated that the discovery learning and learner control of instruction inherent in guided Logo instruction positively affected both near and far transfer of cognitive monitoring strategies. It is recommended that further research be conducted for longer periods of time, using different academic subject material, younger students, and students of differing achievement levels. (68 references) (Author/DB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED336074

Guided Instruction with LOGO Programming and the Development of Cognitive Monitoring Skills among College Students

by Mi Ok C. Lee

Paper presented at the Annual Conference
of the American Educational Research Association
(Chicago, IL, April 3-7, 1991)

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Mi Ok C. Lee

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR015083

BEST COPY AVAILABLE

Guided instruction with Logo programming and the development of cognitive monitoring strategies among college students

Abstract

Educators are facing the fundamental issue of teaching transferrable skills to help students become independent learners and problem solvers in order to keep up with a rapidly changing technological information society. Cognitive monitoring has been considered as teachable, decontextual, and transferrable higher order thinking strategy. It has been regarded as one of the most important and essential strategies for efficient thinking and problem solving.

Logo programming has been advocated as a powerful tool to help students develop higher order thinking, an awareness of their thinking processes, and problem solving skills. There is, however, conflicting evidence for developing thinking and problem solving skills through Logo programming. Recently it has been suggested that Logo programming alone is not enough to guide students toward the development of higher level cognitive skills.

This research examined the effects of guided instruction with Logo programming on the development of cognitive monitoring strategies among college students. Guided Logo instruction in this study involved three pedagogical elements. First, Logo programming was selected as a particular learning tool to teach cognitive monitoring strategies. Second, an explicit instructional model of cognitive monitoring processes was included in the Logo programming. The model specifically emphasized the following cognitive monitoring processes: decomposing, planning, executing, identifying errors, and debugging. Third, explicit teacher mediated instruction of cognitive monitoring activities was facilitated in a Logo programming environment as well as outside of the programming domain.

The results of this study demonstrated positive effects of guided Logo instruction on both the near transfer and far transfer of cognitive monitoring strategies. The result also indicated that guided instruction with Logo programming contributed most significantly to the development of Logo error identification and debugging skills.

Guided instruction with Logo programming and the development of cognitive monitoring strategies among college students

Mi Ok C. Lee
Ann D. Thompson

Iowa State University

Introduction

As our society changes and becomes more complex, and as new knowledge and technology emerge, the need for teaching students independent thinking and problem solving skills is increasing. Responding to the ever increasing societal demands, educators continue to search for proper learning tools and methodologies to improve students' higher-order thinking and problem solving skills (National Commission on Excellence in Education, 1983; Smith, 1987; Task Force on Teaching as a Profession, 1986).

In particular, cognitive monitoring has been a primary concern of educators and cognitive psychologists as research indicates that cognitive monitoring is an important strategy for efficient thinking and problem solving. The term cognitive monitoring is relatively new, but the basic concept has been widely accepted in most areas of research. Research indicates that there is a considerable difference between the abilities of good and poor problem solvers in the use of cognitive monitoring strategies. Good problem solvers more often use cognitive monitoring strategies such as defining a problem, predicting an outcome, planning ahead, regulating the process, checking errors, and modifying their thinking processes (Derry, 1989; Wagner & Sternberg, 1987).

However, many of today's college students and young American adults lack the cognitive monitoring strategies necessary to become independent problem solvers and efficient thinkers. Many college students have great difficulty managing and evaluating their own learning efforts (Chipman & Segal, 1985; Baker, 1989; Shoenfeld, 1985). Although there is a great need to improve cognitive monitoring strategies, there has been little research on proper learning tools and

instructional methods that can facilitate the development of a student's cognitive monitoring strategies. More empirical research is needed to test the potential effects of various instructional tools and methods used to develop cognitive monitoring strategies. One learning tool that has been advocated as a potential to improve problem solving strategies is Logo programming. However, there is very little empirical research available that tests the effects of using Logo programming as a learning tool to improve students' cognitive monitoring strategies.

Thus, purpose of this study was to empirically examine the effectiveness of guided instruction with Logo programming on the development of cognitive monitoring strategies among college students. This study focused on two transfer effects of guided instruction with Logo programming. First, the near transfer effect of such instruction on the development of cognitive monitoring strategies in solving Logo programming problems was examined. Second, the far transfer effect of guided instruction with Logo programming on the development of cognitive monitoring strategies in solving everyday problems was studied.

Theoretical Background

Cognitive monitoring

Cognitive monitoring is a process of regulating or evaluating one's on- going cognitive process while solving problems (Brown, 1987; Flavell, 1976; Forrest-Pressley & Waller, 1984; Van Haneghan & Baker, 1989). Components of cognitive monitoring, in general, are identifying problems, predicting outcomes, planning an efficient solution, self-testing, and assessing the outcome. These cognitive monitoring strategies can be applied for most subjects and in many different problem situations (Brown, 1978, 1987; Derry, 1989). They have been regarded as one of the most important and essential strategies for efficient problem solving (Brown, 1978; Ford, 1981; Lochhead, 1988).

Research on adult metacognition suggest that cognitive monitoring activities play a vital role in task performance. Poor performers do not monitor their performance or modify their strategies in ways that are appropriate for changing task situations. Good adult readers plan strategies,

monitor an on-going performance, revise their previous strategies, and thus perform at a high level (Bereiter and Bird, 1985; Lundeberg, 1987; Wagner & Sternberg, 1987). These studies indicated that high-achieving students and low-achieving students differed greatly in the use of cognitive monitoring strategies, such as seeking information, planning ahead, evaluating the process, and modifying strategies. Thus, high-achieving students were more likely to activate cognitive monitoring than were low-achieving students (Wagner & Sternberg, 1987; Zimmerman and Pons, 1986). Cognitive monitoring strategies are vital skills for problem solving and critical to explaining individual differences in intelligence.

Despite the value of cognitive monitoring, research suggests that many college students and adults lack the necessary cognitive monitoring strategies to maximize learning endeavors and solve problems efficiently (Schoenfeld, 1985; Simpson, 1984; Sternberg, 1986). Thus, a need exists to teach cognitive monitoring strategies to college students and adults in order to help them become efficient thinkers and independent problem solvers in varied situations.

Components of cognitive monitoring strategies selected for this study were decomposing, planning, executing, identifying errors, and debugging errors. These cognitive monitoring activities are recursive. The process of cognitive monitoring activities continues and recycles until students achieve the desired goals.

Logo programming

Logo is a programming language developed by Seymour Papert and his colleagues. The pedagogy of Logo is that children learn through self-guided discovery methods. Proponents of Logo programming believe that Logo has innate educative properties and provides a special learning environment for students to develop learning strategies and problem solving skills (Ryba & Chapman, 1983; Tractenberg, 1985; Watt, 1982). It has been claimed that Logo programming facilitates a learning environment in which students can practice cognitive monitoring activities (Clements, 1990; Harvey, 1982; Papert, 1980). Papert (1980) pointed out that "in teaching the

computer how to think, children embark on an exploration about how they themselves think" (p. 19).

It has been suggested that potential as a powerful tool to develop self-monitoring since Logo environments allow students to engage in dynamic activities, such as selecting their own goals, outlining solution, and testing hypotheses (e.g., Clements, 1987, 1990; Miller & Emihovich, 1986; Papert, 1980). Immediate and explicit error messages in Logo also motivate students to debug their errors. The nature of the Logo environment allows students to consciously reflect and monitor their thinking processes; thus, Logo programming appears to be a logical and appropriate tool for students to explicitly practice cognitive monitoring.

Development of cognitive monitoring through Logo programming

Logo programming has been cited as a powerful vehicle for teaching students about their own thinking processes through a self-discovery approach (Papert, 1980). Papert's philosophy of learning Logo is based on a self-discovery method that uses minimal adult guidance. Papert argued that the Logo environment is an artificial, miniature reality that allows students to explore their ideas, and manipulate and test hypotheses through self-discovery. According to Papert, through Logo activities, students interact socially with groups solving problems together and develop intellectual growth. Most importantly, he suggested that through the self-discovery learning environment in Logo programming, students sharpen their thinking and monitor their thinking process while working with the turtle.

There is, however, conflicting evidence for Papert's claim (e.g. Pea & Kurland, 1984; Salomon & Perkins, 1987). Recently it has been suggested that Logo programming alone is not enough to guide students toward the development of higher level cognitive skills. It has been indicated that Logo programming combined with guided instruction is needed to help students acquire more than just technical programming skills (Seidman, 1987). The guided instruction approach to teaching computer programming has been discussed as a method to encourage the transfer of problem solving skills to other domains. A number of studies examining the self-

discovery approach to learning Logo programming indicated that many students have difficulties in developing problem solving strategies through the self-discovery (Fay & Mayer, 1987; Kurland & Pea, 1985; Pea, 1983; Pea & Kurland, 1984; Perkins, 1985; Swan & Black, 1989).

In guided instruction, the teacher assigns more responsibility to students as they become more capable of performing a complex task. Eventually, the students take the initiative in learning and the teacher becomes a facilitator to guide them only when needed. In this manner, students gradually control their on-going cognitive activities. Thus, guided instruction can assist students to focus on the problem, to search systematically for information, to compare, to develop an insight, to monitor, and to evaluate their on-going activities carefully (Feuerstein, 1979; Missiuna, Hunter, Kemp, & Hyslop, 1987; Samuels, 1986).

Although, there is a support for guided Logo instruction to develop problem solving strategies (e.g., Bamberger, 1984; Seidman, 1987; Swan & Black, 1989), there is no powerful empirical research support for the use of Logo to develop cognitive monitoring strategies. From the past studies on Logo programming, two important research questions are derived: 1) Does learning Logo programming affect a student's cognitive skills? and 2) Does the instructional method for learning Logo programming affect a student's cognitive skills?

Logo has potential as a powerful tool to activate a student's cognitive monitoring strategies since Logo environments allow students to engage in dynamic activities, such as selecting their own goals, outlining solution, and testing hypotheses. Immediate and explicit error messages in Logo also motivate students to debug their errors. Can the nature of Logo programming learning environments enhance a student's cognitive monitoring strategies through self-discovery? Or, do students need explicit instruction in cognitive monitoring to maximize the potential of Logo programming to improve their cognitive monitoring strategies? Since cognitive monitoring strategies are considered as essential for efficient problem solving, searching for a proper learning tool and instructional method to enhance cognitive monitoring strategies is necessary.

Guided Logo instruction employed in this study consisted of three pedagogical elements: 1) Logo programming was selected as a particular tool to teach cognitive monitoring strategies, 2)

explicit components of cognitive monitoring was modeled in Logo instruction, and 3) teacher mediated learning that guides students to apply steps of cognitive monitoring activities in order to solve Logo programming problems as well as outside of Logo programming problems.

Method

Subject

The subjects were 132 students enrolled in class of Secondary Education 101 in the Fall, 1990 at Iowa State University. This class entitled "Educational Applications of Computers", was a three credit elective course for pre-service teachers. At the beginning of the course, a questionnaire was given to the subjects in order to obtain academic and demographic data on each student. From the academic and demographic data, the following seven variables were used as covariates: gender, year in college, mathematics courses taken in high school, computer courses taken in either high school or college, computer ownership, computer confidence, and students' ACT score. The study employed a one factor analysis of covariance design. ACT scores, obtained from the Registrar's office, were used as the pre-experimental measurement of students' academic performance. ACT means for each of the nine laboratory sections were calculated. Then, students were randomly assigned to either the experimental or control treatment by laboratory means of their ACT scores.

As a result, 65 students were in the experimental group and 67 were in the control group. No significant differences existed between the experimental and control groups on the mean of ACT scores. During the experimental period, two students dropped the course because of personal reasons, one from each treatment. Therefore, 130 subjects completed the entire study. The experimental study lasted four weeks, total of 16 hour instruction plus minimum 8 hours of homework assignment.

Experimental procedure

Treatment of guided Logo instruction was given to the students in the experimental group while the self-discovery learning was employed for the students in the control group. Students in

the experimental received Logo programming instruction that systematically guided students through formal cognitive monitoring activities while solving Logo problems; students practiced each of the following: decomposing, planning, executing, identifying errors, and debugging errors. Both the experimental and control groups received the same Logo instructional content and worked on the same set of in-class problems and examples. Both groups used activity worksheets. The experimental group received worksheets for formal cognitive monitoring activities connected with Logo programming problems.

The worksheets consist of five major activities for cognitive monitoring (Appendix A). The decomposing activity was designed to help students search for definite and necessary elements of a complex graphic problem. The planning activity was designed to help students organize and sequence the decomposed information and elements. Most important, students were encouraged to search for strategies in order to develop an efficient solution to the problem. The executing step was designed to help students translate the planned solution into Logo programming commands and execute them on the computer. In the process of identifying the problem, students were encouraged to describe a discrepancy between the desired graphic and the actual outcome, locate errors, and explain what was wrong, if any errors exist. This step was designed to help prevent students from mindlessly finding errors through trial and error without understanding the causes for errors. In the debugging step, students were asked to correct errors.

Following these steps, students pursued cognitive monitoring activities recursively until they solved a problem. Through the recursive, non-linear cognitive monitoring activities, students reformated their original strategy or method and refined the solution. In the process of presenting the cognitive monitoring activities, an instructor intentionally raised a Socratic dialogue to help students concentrate on formal cognitive monitoring activities. With the instructor's guidance, students worked through the steps of formal cognitive monitoring activities on worksheets which were designed to elicit those cognitive monitoring strategies. Finally, the instructor presented general problem situations from various subject areas and guided students to utilize learned cognitive monitoring processes.

The control group received instruction based on a self-discovery learning approach. In contrast to the experimental group, students in the control group did not focus explicitly on the formal cognitive monitoring process while solving the Logo problems. However, the lesson outline was carefully designed in order to deliver the same information given to the experimental group. First, the instructor introduced rationale, new concepts, and commands for the Logo instruction unit. Then, students worked individually on activity sheets in order to solve the given programming problems. Students in the control group received a large degree of freedom when they solved Logo programming problems. Finally, the instructor gathered one or more different answers from students and presented the solution to the entire class. If the answer produced errors, then students debugged the errors by themselves without the instructor's help. Thus, students in the control group spent more time programming according to their own learning style and were able to compare several different solutions. The control group received worksheets that contained Polya's (1973) four steps of problem solving (Appendix B). Such control group worksheets implicitly provided general guidelines in order to prevent students from mindlessly engaging in problem solving. However, this group did not receive explicit formal cognitive monitoring instruction. Students were allowed a large degree of freedom in selecting various strategies for solving a problem according to their own learning styles.

Test instruments

After four weeks of intensive Logo instruction, both groups were given five tests in order to determine the transfer effects. Two tests were used to measure the near transfer effect: 1) the Logo decomposing and planning test and 2) the Logo error identification test. The following three tests were used to determine the far transfer effect: 1) the general decomposing test, 2) the general planning test, and 3) the general error identification test. These tests involved non-Logo problems.

Few test instruments existed that measure the transfer effects of cognitive monitoring strategies. Thus, the researcher developed the test instruments. In order to develop the instruments, the researcher conducted a literature review in each of the subject areas (e.g.,

Bransford & Stein, 1984; Burton, 1984; Clements, 1986, 1987, 1990; Cory & Walker, 1986; Dulit, 1975; Goldin & Hays-Roth, 1980; Hyde, 1989; Lehrer, 1989; Robek, 1989). Then using telephone discussion, the researcher consulted some of the authors to develop test items in the context of this research theme. After the consultation, some of authors provided the researcher with detailed research materials relevant to this work. Following this consultation and a review of the provided materials, the researcher also had discussions on potential test items with professors in computer science, psychology, mathematics, and teacher education. Then, the test items were selected and generated through the discussion of professors in the fields. The content validity of the test items was conducted by the experts in computer programming, mathematics, cognitive psychology, and teacher education. After several informal pilot studies, the researcher modified and refined the tests. Then, the refined tests were again piloted with the students in Secondary Education 101, in Spring and Summer, 1990 (the same course used in the main study in Fall, 1990).

Logo decomposing and planning test: The purpose of this test was to evaluate students' abilities to break a complex Logo graphic into simpler, elementary shapes and to devise a modular and efficient solution to the programming problem. This test contained five different items from simple problems to difficult ones. Each problem provided a graphic which had not been used in the class activity and and student work was scored first for the decomposing skills and second for the planning skills.

There were three questions for each problem. The first question (part A) was to measure a student's decomposing skills. Students were asked to break down the given graphic and list the names of subprocedures they would use in a program to create the graphic. First, the absolute value was calculated by subtracting the number of a student's subprocedures from the optimal number of decomposed subprocedures. Then, the absolute value was subtracted from possible points for this part. The second question (part B) was to measure a student's ability to plan an efficient solution to the problem. Students needed to think not only about the efficient turtle trip but also the logic and modularity of the planning to solve the problem. Possible total points for

this part was five points. Criteria for the optimal planning solution was the logic in thinking, error occuring in that planning, and modularity of planning. In the third question (part C), students were requested to write Logo code consistent with their solution. The third question was used to help an instructor better understand a student's planning strategies. Four criteria were encountered in scoring: Including necessary statements in each procedure, necessary subprocedures, and calling procedure.

Logo error identification test: The purpose of this test was to evaluate a student's error identification and debugging skills in Logo. The test contained six problems. Each problem presented two graphics: the planned graphic and the actual outcome. Each problem also included codes of procedures for the actual outcome.

Students were asked four specific questions for each problem: A) describe the discrepancy between the planned graphic and the actual outcome; B) circle the specific statement(s) that caused the critical error; C) describe what is wrong with the statement(s) circled; and D) write the correct code(s) for the circled statement(s). Each part was scored separately. Some problems had more than one mistake and had more than one possible solution. Therefore, student responses varied. However, re-writing all codes over again was considered as wrong answer and no point was given.

General decomposing test: The purpose of this test was to evaluate students' ability to break a complex problem into manageable parts and correctly construct the solution using the decomposed parts. The measurement was designed to test if students were able to apply principles of Logo decomposing skills to general complex problems. The test contained six word problems combining mathematical with non-mathematical items. For each problem, students needed to break down the given problem into simpler parts, solve decomposed steps, and give a final answer to the problem. For the scoring criteria, the number of decomposing steps and the correctness of the decomposed problem solution were scored separately. Each correctly decomposed step gained a point and each correct answer to the decomposed problem solution also gained a point. However,

if students obtained an incorrect answer at an earlier step and used the same answer consistently, she lost a point only at once.

General planning test: The purpose of this test was to measure a student's ability to organize given information, to order a sequence of actions to be performed, and to think logically in order to achieve the goal most efficiently. The test contained two complex problems. The first problem was about meal planning. Students were asked to finish six different tasks to prepare a dinner. The time constraint was given in this problem statement. The computer program calculated the differences between the optimal cooking time and student cooking time for each task. The second problem was about shopping planning. In this problem, students were required to think not only about the limited given time to do shopping, but also about the sequence of important tasks to be done prior to other tasks (e.g., need to sign paper at the lawyer's office since it is due today), and content of shopping (e.g., buy the ice cream at the end of shopping). Since the shopping map provided several stores, students needed to plan an efficient route for shopping in order to maximize task performance within the limited time. The score for each subject was calculated by a computer program. The computer program was written by C language. Once the student shopping place and shopping time, the program calculated difference between optimal shopping time for each task and compute the total time difference from the given shopping time.

General error identification test: The purpose of this test was to evaluate students' ability to detect what was wrong with a given problem, find errors in a given solution process, explain the problem in the errors, and correct the indicated errors. The measurement was to test if students were able to apply monitoring strategies to other domain problems. The test contained six word problems combined with mathematical and non-mathematical items. Each problem presented steps of the solution process and the final answer. Then students were asked four specific questions for each problem: A) whether the given solution process and answer to the problem was correct or not; B) if the solution was not correct, then circle the specific statement(s) that caused the critical error; C) describe what was wrong with the statement(s) circled; and D) write the correct answer. Each part was scored separately and added for total score.

Cronbach's alpha was applied to measure the internal reliability of the test instruments: results were .79 for Logo planning and decomposing test (15 items); .79 for Logo error identification test (12 items); .74 for the general decomposing test (12 items); .65 for general planning (8 items); and .75 for general error identification test (12 items). The researcher and one other instructor scored the collected tests individually in order to test the reliability of the scoring system. The correlation coefficient for scoring-rescoring reliability were $r=.97$ ($p<.001$) for Logo decomposing and planning test, $r=.97$ ($p<.001$) for the Logo error identification test, $r=.98$ ($p<.001$) for the general decomposing test, $r=.94$ ($p<.001$) for the general planning test, and $r=.97$ ($p<.001$) for the general error identification test.

Results

Statistical analysis of the pre-experimental measures are reported. Based on the statistical analysis of student t-test and nonparametric Chi-square test, the researcher noted that there were no statistically significant differences between the two groups on any of the following background characteristics: 1) demographic background: gender, year in college, accessibility of their own computer, and computer confidence, and 2) academic background: mathematics courses taken in high school, computer courses taken either in high school or college, and ACT scores.

The independent variable was the instructional methodology to teach Logo programming: guided instruction vs. self-discovery instruction. The six dependent variables for the study were Logo decomposing skills, Logo planning skills, Logo error identification skills, general decomposing skills, general planning skills, and general error identification skills. Seven variables were used as covariates in the analysis of the hypotheses: gender, year in college, mathematics courses taken in high school, computer courses taken in either high school or college, computer ownership, computer confidence, and ACT scores.

A multiple analysis of covariance (MANCOVA) test was used to test the treatment effect for the combined dependent variables. The result of Wilks multivariate test revealed that there were statistically significant differences in the vector of means for the experimental group and the control

group, F (6, 111) = 17.78, p<.001. Following this test, an analysis of covariance (ANCOVA) test was used to test if there were difference between the two groups on each of the dependent variables.

Value	Exact F	Hypoth. DF	Error DF	Signif. of F
.510	17.78	6	111	.001***

Logo decomposing skills: The result showed that there was no significant main effect of the treatment on the Logo decomposing test, F(1, 116) = 2.76, p<.099.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	67.36	7	9.62	.75	.631
Main Effects					
Treatment	35.42	1	35.42	2.76	.099
Explained	102.78	8	12.85	1.00	.439
Residual	1489.26	116	12.85		
Total	1592.03	124			

Logo planning skills: The result revealed that there was a statistically significant difference between two groups on the Logo planning test, F(1, 116) = 18.77, p<.001. Among the seven covariates, students' ACT score contributed significantly to the reduction of within group variance, F (1, 116) = 6.77, p<.011.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	1865.52	7	266.50	2.75	.011*
Main Effects					
Treatment	1821.62	1	1821.62	18.77	.001***
Explained	3687.14	8	460.89	4.75	.001
Residual	11259.79	116	97.07		
Total	14946.93	124	120.54		

Logo error identification skills : The result showed that there was a statistically significant difference between the two groups on the Logo error identification test, F(1, 116) = 100.86, p<.001. Two control variables were significant contributors to the reduction of error variance: computer ownership, F(1, 116) = 6.30, p<.013 and ACT scores, F(1, 116) = 23.57, p<.001.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	1435.85	7	205.12	6.61	.001***
Main Effects					
Treatment	3128.84	1	3128.84	100.86	.001***
Explained	4564.69	8	570.59	18.39	.001
Residual	11259.79	116	97.07		
Total	8163.09	124	65.83		

General decomposing skills: The result indicated that there was a statistically significant difference between the two groups on the general decomposing test, $F(1, 116) = 24.96$, $p<.001$. Again, the covariate of ACT score was a significant contributor to the reduction of error variance, $F(1, 116) = 24.96$, $p<.001$.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	2778.68	7	396.96	4.44	.001***
Main Effects					
Treatment	2232.14	1	2232.14	24.96	.001***
Explained	5010.82	8	626.35	7.00	.001
Residual	10375.98	116	89.45		
Total	15386.80	124	124.09		

General planning skills: The result revealed that there was a statistically significant difference between the two groups on the general planning test, $F(1, 116) = 5.41$, $p<.022$. The ACT scores contributed significantly to the reduction of error variance, $F(1, 116) = 7.82$, $p<.006$.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	307.68	7	43.96	2.22	.037*
Main Effects					
Treatment	107.12	1	107.12	5.41	.022*
Explained	414.80	8	51.85	2.62	.011
Residual	2295.32	116	19.79		
Total	2710.12	124	21.86		

General error identification skills: The result showed that there was a statistically significant difference between the two groups on the general error identification test, $F(1, 116) = 26.88$, $p<.001$. The control variable of ACT score was significant, $F(1, 116) = 17.77$, $p<.001$.

Source of variation	Sum of Square	df	Mean Square	F	Signif. of F
Covariates	1106.39	7	158.06	3.32	.003**
Main Effects					
Treatment	1279.62	1	1279.62	26.88	.001***
Explained	2386.01	8	298.25	6.27	.001
Residual	5522.34	116	47.61		
Total	7908.35	124	63.78		

Summary of results

Statistical analysis of the pre-experimental measures indicated that no statistically significant differences were found between the experimental and the control groups on the students' academic and demographic backgrounds.

The results relating to the six dependent variables were reported. The result on combined means for the dependent variables indicated that there was a statistically significant difference between the two groups on the combined means for the six dependent variables. The ANCOVA results indicated that students in the guided Logo instruction group performed significantly higher than students in self-discovery instruction group on the measurements of the following skills: Logo planning skills, Logo error identification skills, general decomposing skills, general planning skills, and general error identification skills. There was no statistically difference between the two groups on the Logo decomposing test.

Discussion

Discussion of overall transfer effect results

The combined means of the dependent variables were significantly different between the guided instruction group and the self-discovery group in Logo programming. Since the combined dependent variables included both the near and far transfer effects of guided instruction, the significant result indicates that the treatment influenced the acquisition of cognitive monitoring strategies in Logo programming as well as outside of programming domains. This finding supports previous Logo research suggesting that guided instruction with Logo programming could facilitate the development of specifically targeted strategies (Grandgenett, 1989; Seidman, 1987; Swan & Black, 1989). This finding also contributes to the research suggesting that guided

instruction focusing on specific strategies helps students acquire and transfer those strategies to their learning in other domains (Brown, 1983; Brown, Campione, & Day, 1981; Palinscar & Brown, 1989). Thus, this result suggests that guided instruction with Logo programming targeting cognitive monitoring strategies will make a positive difference in the development of students' cognitive monitoring strategies.

Discussion of the near transfer effects

Overall, the positive near transfer effects for the cognitive monitoring strategies support the use of guided Logo instruction. The students in the guided Logo group were encouraged to follow steps of formal cognitive monitoring activities to solve in-class Logo problems as well as to complete homework assignments. Students in the guided Logo instruction spent much less time on programming experience than the self-discovery group since Logo was used as a tool to practice particular problem solving strategies. Nevertheless, this group performed better on the measurement of Logo planning and Logo error identification skills. This study supports the previous research suggesting that a self-discovery approach to Logo programming does not seem to develop planning skills, debugging skills, and problem solving skills (Carver, 1987; Leron, 1985; Mayer, 1988; Pea & Kurland, 1984). The results from this study, combined with the others, suggest that a guided instructional approach to teaching cognitive monitoring strategies through Logo programming is very useful to achieve near transfer effects. Thus, positive results on the near transfer effect of guided instruction with Logo programming significantly contribute to the body of research knowledge on the importance of Logo instructional methodology.

Discussion of the far transfer effects

Although some researchers agreed that experiencing computer programming itself enhances higher order thinking skills and general problem solving skills (e.g., Green & Jaeger, 1983; Lawler, 1985; Papert, 1980; Watt, 1982), the results from this study do not support these claims. The results strongly support the body of research knowledge that an explicit instructional model of particular problem solving skills along with teacher mediated learning produces the positive

transfer of those problem solving skills (e.g., Derry, 1989; Grandgenett, 1989; Palinscar & Brown, 1984, 1989; Scheonfeld, 1985; Swan & Black, 1989; Van Haneghan & Baker, 1989).

The positive results of far transfer effects in this study also significantly contribute to the theoretical research on the high road transfer mechanism. When students were trained to utilize decontextualized strategies and principles deliberately and effortfully, they are likely to transfer them to any other learning domains even though the strategies and principles were not automatized (e.g., Salomon & Gardner, 1987; Salomon & Globerson, 1987; Vygotsky, 1978). The researcher did not expect that students would master cognitive monitoring strategies automatically from four weeks of guided instruction with Logo programming. However, the research clearly supported the concept that students were able to use the principles of these strategies in solving problems within the Logo domains as well as outside of the programming domains. The results suggest that the guided Logo instruction environment is necessary in order to help students develop specific problem solving strategies while learning Logo programming.

Recommendations

Based on the findings of the study, the following recommendations for further research emerge:

1. The treatment period in this study was relatively short. A study over a longer period of time seems warranted from the positive findings of this study. A longer study might reveal different effects for each group.

2. It is possible that an explicit instructional model of cognitive monitoring and the nature of mediated learning can be constructed in other subject matters and can be utilized as effectively as the Logo computing environment to teach cognitive monitoring strategies. Further investigation of these possibilities could yield alternative learning tools to teach higher order thinking strategies.

3. Structured Logo instruction may limit students' degree of freedom to employ varied problem solving strategies according to their learning style. Some students may do better when a

large degree of freedom is given in their learning. It is valuable to examine the congruence students' learning style and instructional method.

3. This research focused on college students in order to examine the effects of guided Logo instruction. Younger students may respond differently to the guided Logo instruction. The positive results from this study suggest the need of further research on guided instruction with Logo programming and the development of cognitive monitoring strategies among various age levels and grade levels.

4. Some studies indicated that students learn differently according to their achievement level. Low achievers might learn better through systematically guided instruction whereas high achievers might learn better through self-discovery learning. However, this issue was not addressed in this study. Further studies should examine the influence of student achievement level and other learner characteristics on the treatment effects.

Conclusion

This study focused on guided Logo instruction and the development of cognitive monitoring strategies. The study was conducted to provide empirical evidence for the conflicting results on the effects of instructional methodology in teaching problem solving through Logo programming. The study also clearly supported the power of guided instruction with Logo programming to develop cognitive monitoring strategies. Specifically, Logo programming appears to be a powerful tool to provide dynamic learning environments where students can develop abstract thinking skills. This study indicated that guided instruction facilitated the potential of Logo programming to provide students with cognitive monitoring activities.

Educators are facing the fundamental issue of teaching transferrable skills to help students become independent learners and problem solvers in order to keep up with a rapidly changing technological information society. Cognitive monitoring has been considered as teachable, decontextual, and transferrable higher order thinking strategy. It has been regarded as one of the most important and essential strategies for efficient thinking and problem solving. This study

demonstrated a successful instructional technique for developing cognitive monitoring strategies and the results suggest directions for future work in this area.

References

Anderson, J. R., & Reiser, B. J. (1985). The LISP tutor. Byte, 10, 159-175.

Baker, L. (1989). Metacognition, Comprehension Monitoring, and the Adult Reader. Educational Psychology Review, 1(1), 3-38.

Bamberger, H. J. (1984). The effect of Logo (Turtle Graphics) on the problem solving strategies used by fourth grade children. Unpublished doctoral dissertation. The University of Maryland, ML.

Bereiter, C., & Bird, M. (1985). Use of thinking aloud in identification and teaching of reading comprehension strategies. Cognition and Instruction, 2, 131-156.

Bransford, J. D., & Stein, B. S. (1984). The IDEAL problem solver: A guide for improving thinking, learning, and creativity. New York, NY: W. H. Freeman and Company.

Brown, A. L. (1978). Knowing when, where, and how to remember: A problem of metacognition. In R. Glaser (Ed.), Advances in instructional psychology (Vol. 1). Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A. L. (1983). Learning to learn how to read. In J. Langer and T. Smith Burke (Eds.), Reader meets author, bridging the gap: A psycholinguistic and social linguistic perspective. Newark, NJ: Dell.

Brown, A. (1987). Metacognition, executive control, self-regulation,and other more mysterious mechanisms. In F. E. Weinert & R. H. Kluwe (Eds.), Metacognition, motivation, and understanding. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brown, A. L., Campione, J. C., & Day, J. D. (1981). Learning to learn: On training students to learn from text. Educational Researcher, 10, 14-21.

Burton, L. (1984). Mathematical thinking: the struggle for meaning. Journal for Research in Mathematics Education, 15, 35-49.

Carver, S. M. (1987). Transfer of Logo debugging skill: analysis, instruction, and assessment. Unpublished doctoral dissertation. Pittsburgh, PA: Carnegie-Mellon University.

Chipman, S. F., & Segal, J. W. (1985). Higher cognitive goals for education. In J. W. Segal, S. F. Chipman, and R. Glaser (Eds.), Thinking and Learning Skills, Vol. 2: Current Research and Open Questions. Hillsdale, NJ: Erlbaum.

Clements, D. H. (1986). Logo and cognition: A theoretical foundation. Computers in Human Behavior, 2, 95-110.

Clements, D. H. (1987). Longitudinal study of the effects of Logo programming on cognitive abilities and achievement. Journal of Educational Computing Research, 3(1), 73-94.

Clements, D. H. (1990). Metacomponential Development in a Logo Programming Environment. Journal of Educational Psychology, 82(1), 141-149.

Cory, S., & Walker, M. (1986). Logo Works: Lessons in Logo. Cambridge, MA: Terrapin Logo.

Derry, S. J. (1989). Strategy and expertise in solving word problems. In C. B. McCormick, G. E. Miller, M. Pressley (Eds.), Cognitive strategy research: From basic research to educational applications. New York, NY: Springer-Verlag.

Dulit, E. (1975). Adolescent thinking a la Piaget: The formal stage. In Robert Grinder (ed.), Studies in Adolescences: A book of readings in adolescent development. New York, NY: Macmillan Publishing.

Emihovich, C., & Miller, G. E. (1986). Verbal mediation in Logo instruction: Learning from a Vygotskian perspective. Paper presented at the AERA. ERIC ED 276 411.

Fay, A. L., & Mayer, R. E. (1987). Children's naive conceptions and confusions about LOGO graphics commands. Journal of Educational Psychology, 79, 254-268.

Feuerstein, R. (1979). The dynamic assessment of retarded performers: The learning potential assessment device, theory, instruments, and techniques. Baltimore, MD: University Park Press.

Feuerstein, R. (1980). Instrumental Enrichment. Baltimore, MD: University Park Press.

Flavell, J. H. (1976). Metacognitive aspects of problem solving. In L. Resnick (Ed.), The nature of intelligence. Hillsdale, NJ: Erlbaum.

Ford, N. (1981). Recent approaches to the study and teaching of "effective learning" in higher education. Review of Educational Research, 51(3), 345-377.

Forrest-Pressley, D. L., & Waller, T. G. (1984). Cognition, Metacognition, and Reading. New York, NY: Springer-Verlag.

Goldin, S. E., & Hayes-Roth, B. (1980). Individual Differences in Planning Processes. Rand N-1488-ONR. Santa Monica, CA: The Office of Naval Research.

Grandgenett, N. F. (1989). An investigation of the potential of guided Logo programming instruction for use in the development and transfer of analogical reasoning. Unpublished doctoral dissertation. Iowa State University, Ames, IA.

Green, C. & Jaeger, C. (1983). Teacher, Kids, and Logo. Irvine, CA: Educomputer Publishing.

Harvey, B. (1982). Why Logo? Byte, 7(8), 163-171.

Hyde, A. A. (1989). Thinking in context: Teaching cognitive processes across the elementary school curriculum. New York, NY: Longman.

Kurland, D. M., & Pea, R. D. (1985). Children's mental models of recursive Logo programs. Journal of Educational Computing Research, 1, 235-244.

Lawler, R. W. (1985). Computer Experience and Cognitive Development: A Child's Learning in a Computer Culture. New York, NY: Halsted Press.

Lehrer, R. (1989). Computer-assisted strategic instruction. In C. B. McCormick, G. E. Miller, M. Pressley (Eds.), Cognitive strategy research: From basic research to educational applications. New York, NY: Springer-Verlag.

Leron, U. (1985). Logo today: vision and reality. The Computing Teacher, 12(5), 26-32

Lochhead, J. (1988). Toward a scientific practice of mathematics education. Paper presented at the conference "Toward a Scientific Practice of Science Education". Berkeley, CA. ERIC ED 294 747.

Lundeberg, M. A. (1987). Metacognitive aspects of reading comprehension: Studying understanding in legal case analysis. Reading Researcher Quarterly, 22, 407-432.

Mayer, R. E. (1988). Introduction to research on teaching and learning computer programming. In R. E. Mayer (Ed.), Teaching and learning computer programming: Multiple research perspectives. Hillsdale, NJ: Lawrence Erlbaum Associates.

Miller, G. E., & Emihovich, C. (1986). The effects of mediated programming instruction on preschool children's self-monitoring. Journal of Educational Computing Research, 2(3), 283-297.

Missiuna, C., Hunter, J., Kemp, T., & Hyslop, I. (1987). Development and evaluation of the "thinking with Logo curriculum". Technical Report No. 143. Edmonton, Alberta. ERIC ED 287 453.

National Commission on Excellence in Education. (1983). A Nation at Risk: The Imperative for Educational Reform. Washington, DC: United States Department of Education.

Palinscar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. Cognition and Instruction, 1, 117-175.

Palinscar, A. S., & Brown, A. L. (1989). Instruction for self-regulated reading. In L. B. Resnick and L. E. Klopfer, Toward the Thinking Curriculum: Current Cognitive Research. Alexandria, VA: 1989 Yearbook of the Association for Supervision and Curriculum Development.

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York, NY: Basic Books, Inc.

Pea, R. D. (1983). Logo programming and problem solving. Technical Report No. 12. New York, NY: Bank Street College of Education, Center for Children and Technology.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. New Ideas in Psychology, 2, 137-168.

Pea, R. D., & Kurland, D. M. (1987). Logo programming and the development of planning skills. In K. Sheingold and R. D. Pea (Eds.), Mirrors of Minds. Norwood, NJ: Ablex Publishing.

Perkins, D. N. (1985). The fingertip effect: How information-processing technology shapes thinking. Educational Researcher, 14, 11-17.

Polya, G. (1973). How to Solve It. Princeton, NJ: Princeton University Press.

Rebok, G. W. (1989). Plan, Actions, and Transactions in Solving Everyday Problems. In J. D. Sinnott (Ed.), Everyday problem solving: Theory and applications. New York, NY: Praeger Publishers.

Ryba, K. A., & Chapman, J. W. (1983). Toward improving learning strategies and personal adjustment with computers. The Computing Teacher, 11(1), 48-49.

Salomon, G., & Gardner, H. (1986). The computer as educator: Lessons from television research. Educational Researcher, 15(1), 13-19.

Salomon, G., & Globerson, T. (1987). Skill may not be enough: The role of mindfulness in learning and transfer. International Journal of Educational Research, 11, 623-637.

Salomon, G., & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how? Journal of Educational Computing Research, 3(2), 149-169.

Samuels, M. (1986). Instrumental Enrichment with learning disabled students. Paper presented at the International Association for Children with Learning Disabilities Meeting, Edmonton, Alberta.

Schoenfeld, A. H. (1985). Metacognitive and epistemological issues in mathematical understanding. In E. A. Silver (Ed.), Teaching and learning mathematical problem solving: Multiple research perspectives. Hillsdale, NJ: Lawrence Erlbaum Associates.

Seidman, R. H. (1987). Research on teaching and learning computer programming symposium. Paper presented to AERA, Washington, DC. ERIC ED 287 442.

Simpson, M. L. (1984). The status of study strategy instruction: Implications for classroom teachers. Journal of Reading, 28, 136-142.

Singley, M. K., & Anderson, J. R. (1989). The transfer of cognitive skill. Cambridge, MA: Harvard University Press.

Smith, K. A. (1987). Educational engineering: Heuristics for improving learning effectiveness and efficiency. Engineering Education, 77(5), 274-279.

Sternberg, R. J. (1986). Intelligence applied: Understanding and increasing your intellectual skills. San Diego, CA: Harcourt Brace Jovanovich.

Swan, K., & Black, J. B. (1989). Logo Programming and the Teaching and Learning of Problem Solving. CCT Report 89-1, Columbia University, New York, NY.

Task Force on Teaching As a Profession. (1986). A Nation Prepared: Teachers for the 21st Century. Washington, DC: Carnegie Forum on Education and the Economy,

Tractenberg, R. (1985). Logo and the education of the hearing impaired. Computers in the Schools, 2(2/3), 283-297.

Van Haneghan, J. P., & Baker, L. (1989). Cognitive monitoring in mathematics. In C. B. McCormick, G. E. Miller, M. Pressley (Eds.), Cognitive strategy research: From basic research to educational applications. New York, NY: Springer-Verlag.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press.

Wagner, R. K., & Sternberg, R. J. (1987). Executive control in reading comprehension. In B. Britton & S. M. Glynn (Eds.), Executive control processes in reading. Hillsdale, NJ: Lawrence Erlbaum Associates.

Watt, M. (1982). What is Logo? Creative Computing, 8(10), 112-129.

Zimmerman, B. J., & Pons, M. M. (1986). Development of a structured interview of assessing student use of self-regulated learning strategies. American Educational Research Journal, 23, 614-628.

Appendix A: Example of the student activity sheet for the experimental group

Cognitive Monitoring Sheet

Lecture4/A1

Graphic

(Teacher-Student Cooperative Activity)

Decompose:

Plan:

Execute:

Identify error:

Debug:

Directions:

- Apply the steps of cognitive monitoring strategies. *(solution process)*

1. **Decompose:** Break a large graphical output into manageable units.
2. **Plan:** Organize and sequence each component of the decomposed output in order to devise an efficient solution. *(e.g., label each line, turn, move, repeat, or procedure)*
3. **Execute:** Code each sequence into Logo commands.
4. **Identify error:**
   - A. Describe discrepancy between the given graphic and the actual outcome.
   - B. Locate syntax errors. *(e.g., To-end, use of the same procedure name more than once, space between primitives and input, misspelling, etc.)*
   - C. Locate semantic errors. *(e.g., subprocedures are called in wrong order, too many/few repeat processes, turn, move, etc.)*
   - D. Explain the cause(s) of the detected errors.
5. **Debug:** Revise detected errors.

•Evaluation of cognitive monitoring strategies:

Decompose: Right size to handle_____ Too small_____ Too large_____

Plan: Appropriate_____ Not appropriate_____ Not quite appropriate_____

Debug: What seems wrong? (Describe in words.)

Appendix B: Example of the student activity sheet for the control group

Class Activiy Sheet

Lecture3/B2

Graphic

(Student Activity)

Execution

(Write code to draw a graphic output)

Guidance:

1) Sketch the graphical output desired. *(Defining the problem)*

2) Think about what you need to do in order to have the turtle draw the desired graphic shown above. It may be useful to look at some past problems from your notes. *(Searching for useful strategies)*

3) Now build a program to have the turtle draw the desired graphic. You may want to either write codes on the paper first or start programming directly on the computer. *(Carrying out the desired output)*

4) How did the program work? Briefly describe below: *(Looking back)*

Desired Output?	**Part(s) to Modify?**	**Other Ways to Solve the Problem?**
Yes Close No	Describe in words	Describe in words